CASE tools: constructivism and its application to learning and usability of software engineering tools

Author(s): Fowler, Lynne ; Armarego, Jocelyn ; Allen, Maurice

Year: 2001

Source: *Computer Science Education*, vol. 11, iss. 3, pp. 261-271.

Official URL: http://dx.doi.org/10.1076/csed.11.3.261.3835

CASE Tools: Constructivism and its Application to Learning and Usability of Software Engineering Tools

Lynne Fowler, Jocelyn Armarego, Maurice Allen
School of Engineering
Murdoch University Murdoch WA 6150
Tel: (08) 9360 7119 Fax: (08) 9360 7104
{maurice , jocelyn, lynne}@eng.murdoch.edu.au

Abstract

Software Engineering is a new discipline aimed at the improvement of the production of large, quality software systems. Interest in CASE tools has grown because of the important role they play in supporting the software development process. Studies show these complex and sophisticated tools have a positive impact on quality and productivity but they have been slow to be adopted by industry; this is partially explained by the difficulty of learning to use the tool.

The constructivist theory of knowledge and learning views knowledge not as pre-existing, but constructed. Individuals are different and these differences affect how a user performs when using a software package. This first phase of our research examines the learning styles of our students and addresses issues confronting them when using a complex software package; namely Rational Rose, our chosen CASE tool.

Keywords: CASE, Learning styles, Online learning, Constructivist theory.

Software Engineering Case Tools and Learning

Education and learning are ongoing and dynamic. As such, our teaching and learning styles and methodologies must be continually reviewed to respond to developments in technology and to the changing demands of society. In universities, as well as in other educational institutions, online computer resources are seen as essential.

Software Engineering is a relatively new discipline grown from rapidly expanding demands in IT and computing. CASE, Computer Aided Software Engineering, is defined broadly:

> ".. as tools and methods to support an engineering approach to software development at all stages of the process. By 'engineering approach' we mean a well-defined, coordinated and repeatable activity with accepted representations, design rules and standards of quality." (Forte and Norman, 1992).

Students need to master numerous software packages during the course of their studies. This is especially so within Software Engineering and Computing programs. These packages are constantly changing, being updated and replaced, with a commensurate increase in complexity. Many professionals in industry who have to maintain state of the art skills also face similar problems to our students in learning these packages. By looking at CASE tool software, this research aims to address learning issues that can aid our approaches to teaching software packages along with considering transferable skills users need to keep up with the dynamic and developing nature of software change. The CASE tool used in this research is *Rational Rose* (Rational, 1999), a professional package, which is currently gaining increased support and recognition within industry. We need to expose our students to software with similar capabilities to commercial software, as well as meeting our educational requirements. This package achieves these objectives (Fowler and Allen, 1999), but its complexity poses the same problems for us in education as for professionals in industry.

CASE tools

CASE tools support software development, being applicable at some or all stages of the development process, and are now more than ever becoming a key area in Software Engineering. The complete Software Development Life Cycle (SDLC) needs support as software systems become more complex, large and often critical. Software costs each year are increasing and modest improvements in productivity of software would mean significant savings (Fuggetta et al., 1993).

CASE tools are aimed to address the difficulties of developing high quality, complex software on time and to budget. Consequently, CASE was thought to provide the solution and these tools were hailed as the answer to the software crisis (Pressman, 1997). However, "CASE has not been the panacea promised by earlier hype", (Gabel, 1994). Most studies show that CASE tools do positively impact quality and productivity (Iivari, 1996) despite having been slow to be adopted by industry (Holt, 1997; Dutta, Lee and Wassenhove, 1999). Statistics show (Kermerer, 1992) after one-year of introduction that:

70% of case tools are never used
25% are used by only one group
5% are widely used but not to their full capacity.

Cited reasons suggested for CASE tools not having been adopted include:

- CASE tools are complex and difficult
- Case tools require a steep learning curve
- Training on tools is often not available
- Training on tools is often not sought by companies
- Managerial attitudes to the use of tools needs to change
- 50% of industry do not follow a well defined process or methodology
- CASE tools are a sharp step forward for many organisations
- Case tools need a large investment in time, cost and effort
- Organisations often use in-house techniques and CASE tools are not easily modified to individual requirements
- CASE tools are generally expensive
- Tools do not perform as well as expected
- Tools do not cover enough of the SDLC.

The *learnability* and *usability* of CASE tools is therefore of paramount importance and investigating learning issues will help both students in education and professionals within industry.

Learning: the construction of knowledge

Learning is a process of acquiring and synthesising ideas and concepts. The process not only involves obtaining information but also full participation by the learner. No longer are the traditional roles of teacher/student: teacher giving, student accepting, considered the only way to learn or even the best way (Kolb, 1984).

Currently-accepted education theory rejects the behaviourist premise that rote learning, reliant on frequent reinforcement of responses, models the way people acquire knowledge (Fosnot, 1996). While behaviourism embodies one of the key principles of positivism: that knowledge of the world can only evolve from observation of objective facts and phenomena, cognitivists claim that a person's response to stimuli is individual and depends on their cognitive state and the mental processes occurring (Dalgarno, 1996). The acknowledgement of a reciprocal relationship between learning and memory (what we learn is affected by its meaningfulness, that meaning is determined by what is remembered and that memory is affected by what we learn (Winn and Snyder, 1996)) and between knowledge and environment has led to a philosophical shift within educational psychology, from objectivism to constructivism. No single correct model of knowledge is widely accepted but knowledge construction depends on:

- what is already known
- previous experience
- organisation of these experiences
- beliefs that the individual uses to interpret the reality of objects and events encountered

(Bruner, 1962;Vygotsky, 1978; Piaget, 1980).

A debate has arisen between cognitive and social constructivists, based on the relative importance placed on individual construction or socio-cultural effects on learning. An individual's cognitive structures are observed only in context, within a culture, but cultural knowing is also a dynamic interplay of individual interpretations, transformations and constructions (Phillips, 1995).

The consequences of this philosophical shift, and more specifically of the constructivist debate, are not clear for teaching and learning with online resources, with differing views extending from moderate to radical (Dalgarno, 1996).

Our cognitivist approach to research in this area has focussed on the styles of learning that apply to either different categories of learners, or the learning of different categories of material, providing insights into individual differences in learning and performance. The challenge is to identify the successful mental modelling strategies of the learner or to modify the learner's approaches to learning (McLoughlin, 1999).

Effective use of complex packages cannot be achieved by behaviourist based drill and rote learning but necessitates a constructivist approach based on experience and exploration. Software applications, and particularly CASE tool software, can be viewed as microworlds – they provide an abstract environment that models the real world from a software development perspective. *Microworlds*, allow the user to create a construction, which will behave in a way consistent with the concepts being modelled (Papert, 1993; Rieber, 1994). A substantial software package should be designed to be flexible enough to allow the user the opportunity to work within the microworld of the package but allow for variation from individual learners who have different learning styles. The importance of learning styles on the construction of knowledge is therefore paramount, coupled with the support for the individual learning style of the user by the software package.

Social constructivists recognise that people and teachers play an active role in the learning process. How the teacher presents a software package will also impact on the student learning process.

Evaluating Usability and Learnability

Usability and learnability of a software package are serious concerns for the area of Human Machine Interaction. Designers have tried for years to seek more direct interaction with users during both the design phase and post development stages of a system or software package in order to assess the interface and its usability (Nielsen, 1993).

Feedback from surveys, interviews, online suggestion boxes, newsletters and online user groups can provide valuable information on a system and whether it meets its design requirements and how well it has been accepted.

Heuristic evaluation is an informal method of analysis where an evaluator studies the interface looking for usability problems and passing judgement according to their own opinions and experience (Nielsen, 1993). This method has been extensively used, but is heavily reliant on the evaluator and very subjective.

Co-operative evaluation has been widely employed; a user is observed whilst performing a task and asked to think aloud. The main issue here is that the observational data is hard to interpret (Nielsen, 1993). The cost and time factors involved are enormous and finding appropriate users is also a problem.

Automatic support and reliable measurable quantities can reinforce and complement results from these methods (Lecerof and Paterno, 1998). Automating the collection of data would allow greater numbers of tests to be performed in a more cost effective and unbiased way.

The purpose of this research is to perform analysis of software use in a quantifiable manner. For example, a measure of performance often used, within the domain of automotive design, is the number of miles/kilometres per gallon/litre. Our aim therefore is to obtain measurements that can be used to assess the software package environment by tracking the online actions of each user, and therefore, addressing the impact of individual learning styles upon the successful usage of the CASE tool.

Individuals are different and these differences affect how a user interacts with a software package. The variation in success when observing a small group of engineering students perform a difficult two hour, ten step, exercise on our CASE tool was extreme: some students completed the task successfully, one actually stormed out of the room completely frustrated. The most successful students, measuring success as completion of the task, followed the ten steps rigorously. The unsuccessful students tended to want to search around in the package and investigate things for themselves. These anecdotal observations have led us to investigate the learning styles of our students and to consider the different aspects involved in the construction of knowledge.

The initial concept of this research was based on a small but significant investigative study to examine the learning issues surrounding a CASE tool (Fowler and Allen, 1999). These results have now guided our investigation of learning styles to the development of an online analyser to monitor use and navigation within the CASE tool.

We aim therefore to draw together the two sides of the theories of constructivism in order to develop a learning style methodology for learning online and in particular learning CASE tools.

Learning Styles

The first phase is to examine learning issues confronting our software engineering students when using CASE tools. The research will then extend into the software industry where the particular CASE tool is being used. The emphasis is on learning styles and teaching methods rather than on student assessment. This study will aid the acceptance of CASE tools in industry and help bridge the learning gap that these complex tools present.

Whilst there are numerous instruments for assessing learning styles, those advocated by Kolb, *Learning Style Inventory*, (1984), and Soloman and Felder, *Index of Learning Styles*, (1999) are well known, and accepted within education theory (Montgomery, 1995). Both instruments provide an efficient way of analysing our students' learning styles and complement each other on the information they supply.

Kolb views the learning process as a four-stage cycle: concrete experience followed by observation and reflection, which leads to the formation of abstract concepts and generalisations, which leads to hypotheses. The hypothesis can then be tested leading to new experiences and the cycle continuing. The Kolb *Learning Style Inventory* is a simple test based on experiential learning theory. It looks at four stages of the learning process: concrete experience (CE), reflective observation (RO), abstract conceptualisation (AC), and active experimentation (AE). A series of twelve questions are presented, with the user ranking four possible answers for each question. Special care is required when explaining to the clients the ranking process. The users' learning style, (Burns, 1989), can then be identified as either:

- **Accomodator**: *What if?* people. Often start with what they see and feel then plunge in and seek hidden possibilities. They learn by trial an error and self-discovery
- **Diverger** : *Why or why not?* These people study life as it is and reflect on it to seek meaning. They learn by being involved and need to listen and share with others
- **Converger**: *How?* These people start with an idea and try it out, they like to find out how things work and learn by testing theories
- **Assimilator**: *What?* people. These people come up with ideas and then reflect on them. They like to know what the experts think.

Our results build upon our previous studies (Fowler et al., 2000). The learning styles of our engineering students are diverse, and span *diverger, assimilator* and *converger* types compared with the general arts students who are heavily *assimilators*, Table 1, indicating the variety of student types that our programs attract. This result is excellent given the multi-disciplinary nature of our curriculum content but we need to be able to cater for all of these student types. Our staff are showing a greater tendency to *assimilator* and *converger* types; this is in line with Kolb (1984) - that engineering is a good career area for *convergers* and that teaching suits *assimilators*. Students whose learning styles are compatible with the teaching style of a course instructor tend to retain information better, obtain better grades and maintain a greater interest in the course (Felder, 1993). An introduction of online course material removes the reliance upon face-to-face teaching, compensating for the mismatch in learning styles between student and instructor but introduces a reliance on the new medium, which may not suit all learners. We are now developing our online material taking into consideration the range of student learning styles.

The *Index of Learning Styles* (Soloman and Felder, 1999) is an instrument to assess learning preferences on four dimensions*; active/reflective*, *sensing/intuitive*, *visual/verbal*, and *sequential/global*. This instrument consists of forty-four simple questions with a choice between two possible answers.

The results from Table 2 show that:

- 55% of the engineering students learn best actively, yet our teachers are mainly reflective
- 70% of the engineering students are sensors, yet our teachers tend to be intuitive
- 79% of the engineering students are visual, yet traditionally material is presented to them verbally or in written form
- 36 % of the engineering students are global learners, yet teaching is often narrowly focused.

Our results for students are similar to those of Mackenzie, (1998) who surveyed 75 Mechanical Engineering students; although our students are showing a greater tendency to sensory (70% compared with 56% from Mackenzie's studies).

Soloman (email to author, November 2, 1999), has surveyed large volumes of students via her online site and her results show that:

- 80% of all students are active learners
- 55% are sensors and 60% for engineers
- 75% are visual learners
- 60% are sequential learners.

Interestingly, the profile of the general arts and commerce students is very similar to that of the engineering students, Table 2, but the Kolb survey, Table 1, has differentiated more clearly between the learning styles of these two groups. The greater tendency towards *assimilators* for the general arts students is consistent with Kolb's description of *assimilators*, as being less practical and more creative.

Application of Learning Style Results

Currently, we introduce *Rational Rose* by means of an online tutorial, which consists of a ten-step exercise, taking approximately two hours to complete, using all stages of the package to finally generate skeleton code for a small program.

From our results, in Table 2, it can be seen that:

- at least half of our engineering students are active learners and hence using practical hands-on sessions directly supports their learning style. This hands-on approach also supports sensory learners

- 36% of our engineering students are global learners and we are directing them to work sequentially. We now aim to provide an initial 30-minute quiz, which will necessitate students searching through the package and therefore obtaining a global view of the software's capability. This approach should also appeal to the active learner who is keen to see how it works
- the majority of learners are visual and special attention has been given to the look of our teaching material, using colour, movement, and a variety of visual prompts. The package itself is particularly visual with its adherence to the UML modelling language and associated graphical icons
- approximately half of our engineering students are reflective learners. Application of Felder and Silverman's idea - stopping several times in a teaching session to pause for thought and pose a short question for discussion - (Felder and Silverman, 1988) would benefit these students. This technique, in the context of the online exercise, would assist in breaking up the long two-hour session, making the experience more stimulating and rewarding.

A suggestion by Felder (Felder, 1993) is to talk to students about their learning styles and the strengths and weaknesses associated with each style. We have now incorporated a topic into our first year Foundation Unit (a general unit which the majority of students complete) to survey and discuss student learning styles. This then gives the student an awareness of issues surrounding their learning and how to get the best from the courses.

Future phases

The next phase of our research is to develop our analyser to monitor patterns of movement within the CASE tool by coding scripts embedded into *Rational Rose*. This analysis of our microworld will be correlated to the students' learning style.

Continuous surveying of first year learning styles is now in place and future developments will also involve surveying students at the end of their degrees to monitor any changes.

Professionals from industry will be included in the long term study for comparison. Extrapolation of our results will enable us to address the training needs of the software industry and hence aiding the uptake of these essential tools.

Conclusion

Learning is a complex process and, as described by the constructivist paradigm, knowledge is internally constructed by the learner. This paradigm encompasses a collection of different perspectives but acknowledges that learning involves making meaning of experiences and therefore that knowledge constructed by the learner is unique.

By identifying individual student learning styles and monitoring student use of the CASE tool software our research is addressing the following critical questions:

- How is knowledge constructed and is this process dependant upon learning style?

- Do software packages, and in particular CASE tool software, allow for different learners with different learning styles to construct the knowledge necessary to use the package?
- Does the way the teacher presents a software package impact on the student learning process?
- Can students having an understanding of their learning styles construct knowledge more effectively in a learning environment contrary to their individual style?

Many software packages and tools are initially self taught or assessed and they are often perceived to be complex. This adds to the difficulty of learning CASE tools and consequently on their uptake within industry. An awareness of how software professionals and students learn will help the approach to tackling these difficult packages.

This research sets the foundation for future work, to develop an online learning methodology, whereby learner characteristics can be used to establish an environment to support the construction of knowledge in students.

References

Bruner, J. S. (1962). On knowing, essays for the left hand. *Harvard University Press*, Cambridge, (MA).

Burns, S. (1989). There's More Than One Way to Learn. *Australia Wellbeing*, 33 42-44.

Dalgarno, B. (1996). Constructivist computer assisted learning: theory and techniques. *Making New Connections: proceedings of ASCILITE '96* University of Adelaide, 127-148.

Dutta, S., Lee, M., and Wassenhove, L. V. (1999, May/June). Software engineering in europe: A study of best practices. *IEEE Software,* 16(3), 82-89.

Felder, R., (1993). Reaching the second tier: Learning and teaching styles in college science education. *Journal College Science Teaching*, 23(5), 286 – 290.

Felder, R., and Silverman, R. L. (1988, April). Learning and teaching styles in engineering education. *Engineering Education*, 78(8), 674-681.

Forte, G., Norman, R. J. (1992, April). A self-assessment by the software engineering community. *Communications of the ACM*, vol 35, no 4, 28-32.

Fosnot, C. (1996). Constructivism: theory, perspectives and practice. *Teachers College Press*, New York.

Fowler, L., and Allen, M. (1999), CASE tools-A basis for developing an online learning methodology. *AWCSET '99*, Oct 1999.

Fowler, L., Allen, M., Armarego, J. and Mackenzie, J. (2000). Learning styles and CASE tools in Software Engineering. In A. Herrmann and M.M. Kulski (Eds), *Flexible Futures in Tertiary Teaching. Proceedings of the 9th Annual Teaching Learning Forum*, 2-4 February 2000. Perth: Curtin University of Technology. http://cleo.murdoch.edu.au/confs/tlf/tlf2000/fowler.html (Aug. 2000).

Fuggetta, A., Politecnico di Milano and CEFRIEL. (1993). A classification of CASE technology. *Computer*, 26(12), 25-38.

Gabel, D. A. (1994). Technology 1994 software engineering. *IEEE Spectrum*, 31(1), 38-41.

Holt, J.D. (1997, Aug). Current practice in software engineering – A survey. *Computing and Control Engineering Journal*, 8(4), 167-172.

Iivari J. (1996, Oct). Why are CASE tools not used. *Communications of the ACM*, 39(10), 94-103.

Jonassen, D. H. (1996). Handbook of Research for Educational Communications and Technology. *Simon and Schuster Macmillan,* New York.

Kermerer, C.F. (1992). How the learning curve affects CASE tool adoption. *IEEE Software*, 9(3), 23-28.

Kolb, D. A. (1984). Experiential learning experience as the source of learning and development. *Prentice-Hall*.

Lecerof, A., Paterno, F. (1998). Automatic support for Usability and Evaluation. *IEEE Transactions on Software Engineering*, 24(10), Oct.

Mackenzie, J. (1998). Computer applications in chemical engineering. *PhD Thesis*, University of Canterbury, NZ.

McLoughlin, C. (1999). The implications of the research literature on learning styles for the design of instructional material. *Australian Journal of Educational Technology* 15(3), 222-241.

Montgomery, S. M. (1995). Addressing diverse learning styles through the use of multimedia. In Engineering Education for the 21st Century: Proceedings of the 25th Annual Frontiers in Education Conference pp 3a2.13-3a2.21. [online] http://fre.www.ecn.purdue.edu/FrE/asee/fie95. (Sept, 2000).

Nielsen, J. (1993). *Usability Engineering*. Boston: Academic Press.

Papert, S. (1993). Mindstorms, children. *Computers and Powerful Ideas*, Harvester and Wheatsheaf.

Phillips, D. C. (1995). The good, the bad, and the ugly: The many faces of constructivism. *Educational Research*, 24(7), 5-12.

Piaget, J. (1980). The psychogenesis of knowledge and its epistemological significance. *Piattelli-Palmerino*.

Piattelli-Palmerino. (1980). M language and learning. *Harvard University Press*, Cambridge, (MA).

Pressman, R. (1997). *Software Engineering A Practitioners Approach*, Fourth Edition, McGraw Hill.

Rational Software. (1999). [online]. http://www.rational.com/ (Aug, 2000).

Rieber, L. P. (1994). *Computers, Graphics and Learning*, Brown and Benchmark.

Soloman, B., and Felder, R. (1999). Index of learning styles (ILS). [online]. http://www2.ncsu.edu/unity/lockers/users/f/felder/public/ILSpage.html (Aug, 2000).

Vygotsky, L. S. (1978). Mind and society: the development of higher psychological processes. *Harvard University Press Cambridge*, (MA).

Winn, W., Snyder, D. (1996). Cognitive perspectives in psychology. Jonassen, 112-142.

Clients	No. of Clients	Accomodator	Diverger	Assimilator	Converger
Engineering Students 1st year					
Engineering Staff					
General Arts and Commerce Students 1st year					

Table 1. *Learning Style Inventory* Results (Kolb)

Clients	No of Clients	Processing	Perception	Input	Understanding
Engineering students	33	Active 55%	Sensory 70%	Visual 79%	Sequential 64%
		Reflective 45%	Intuitive 30%	Verbal 21%	Global 36%
Engineering Staff	9	Active 11%	Sensory 33%	Visual 67%	Sequential 56%
		Reflective 89%	Intuitive 67%	Verbal 33%	Global 44%
General Arts and Commerce Students	51	Active 51%	Sensory 73%	Visual 78%	Sequential 65%
		Reflective 49%	Intuitive 27%	Verbal 22%	Global 35%

Table 2. *Index of Learning Style* Survey Results (Soloman and Felder)